View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Theseus

Tommi Raitanen

VERKONVALVONTAJÄRJESTELMÄN KEHITTÄMINEN

Tietotekniikan koulutusohjelma

2017

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

VERKONVALVONTAJÄRJESTELMÄN KEHITTÄMINEN

Raitanen, Tommi
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Huhtikuu 2017
Ohjaaja: Aromaa, Juha DI
Sivumäärä: 18
Liitteitä: 2

Asiasanat: tietotekniikka, teleoperaattorit, verkonhallinta, IP-verkko

---

Työssä tutkittiin, voidaanko toimeksiantajan käyttämän Smokeping-ohjelmiston luotaimiin lisätä asetus, jonka avulla ohjelman lähettämien IPv4-pakettien Type of Service-kenttään saataisiin tietyt bitit. Lisäksi tutkittiin, onko ohjelmiston mahdollista piirtää kahta rinnakkaista käyrää yhteen kuvaan. Tämä toiminto helpottaisi vertailua eri QoS-määrittelyjen välillä. Jos tämä osoittautuisi mahdottomaksi, piti tutkia ja esitellä vaihtoehtoisia ohjelmistoja, joissa ToS-bitit saadaan sisällytettyä paketteihin.

Työ aloitettiin, koska toimeksiantaja halusi parempaa tietoa siitä, milloin järjestelmä alkaa hukkaamaan paketteja, joka vaikuttaisi Voice over IP-puheluihin. Yleensä tämä johtuu vikatilanteesta, jolloin toimeksiantaja pystyisi reagoimaan ongelmiin nopeammin.

Toimeksiantaja halusi tutkimuksen tehtävän teoriana, mutta koska työhön haluttiin myös dokumentaatiota oppilaitoksen toivomuksesta, päädyttiin luomaan testiympäristö, jossa työn tuloksia voitaisiin todentaa. Laitteistona toimi kaksi tietokonetta sekä reititin. Käyttöjärjestelmäksi valittiin Linux Mint. Työvaiheet koostuivat teoriasta, testiympäristön konfiguroinnista sekä päätelmien teosta tulosten valossa.

IMPROVEMENT OF NETWORK MONITORING SYSTEM

Raitanen, Tommi
Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences
Degree Programme in Computer Science
April 2017
Supervisor: Aromaa, Juha M.Sc.
Number of pages: 18
Appendices: 2

Keywords: computer science, tele operators, network administration, IP network

---

The purpose of this thesis was to research if it would be possible to add certain bits in the Type of Service-field of an IPv4 packet. This field is included in every IPv4-packet sent by probes in a software called Smokeping. In addition, it was to be determined if there was a way to draw two graphs on one screen for different configurations. This would ease the comparison between the results given by these configurations. If this turned out to be impossible the work would include a section for alternative products.

This work was started by the request of the client who wanted to receive better information about packet loss affecting Voice over IP calls. Usually this would indicate a problem in the network so good results of the work would enable the client to react faster in these situations.

The client wanted the research results only in theory, but the university wished for more documentation so a small test environment was created in which the results could be verified. The equipment included two PC's and one router. Linux Mint was chosen as the operating system. Work phases consist of theory, configuring the test environment and conclusion based on the results.

SISÄLLYS

LYHENNELUETTELO

AF	Assured Forwarding
CoS	Class of Service
DDoS	Distruputed Denial of Service
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IoT	Internet of Things
Ipv4	Internet Protocol versio 4
PHB	Per Hop Behaviours
QoS	Quality of Service
ToS	Type of Service
VOIP	Voice Over IP

1 JOHDANTO

IP-verkon palvelun laadun eli Quality of Service:n tarve kasvaa jatkuvasti. Lähitulevaisuudessa tv-streamien kasvaminen vaatii yhä enemmän suunnittelua ja katkeamattoman lähetyksen aikaansaaminen on pakollista. Priorisoinnin pitää olla kunnossa vikatilanteiden sattuessa ja operaattorit haluaisivatkin, että QoS:n kehittämistä jatkettaisiin. Varsinkin radioverkossa tietyn palvelun taso on vaikea luvata, ja operaattorit toivoisivat QoS:ään enemmän dynaamisuutta, sekä asiakaskohtaisen määrittelyn helpottumista (Raikisto 2017).

Tämä opinnäytetyö on kaksiosainen. Ensimmäisessä osassa käsitellään Quality of Servicen teoriaa. Tässä osassa selvennetään, mikä on Quality of Service, mihin sitä voidaan käyttää, ja miten se näkyy arkisessa verkonvalvonnassa. Sen lisäksi työssä esitellään QoS:n osia ja niiden toimintoja, joihin tämä työ enimmäkseen keskittyy, ja jotka näyttelevät suurta osaa työn onnistumisen kannalta.

Toisessa osassa käsitellään teleoperaattorilta saatua toimeksiantoa. Toimeksiannossa tarkasteltiin, onko yrityksen nykyään käyttämään Smokeping verkonvalvontaohjelmistoon mahdollista määrittää Type of Service-bitit jotka ovat osa IPv4-paketteja.

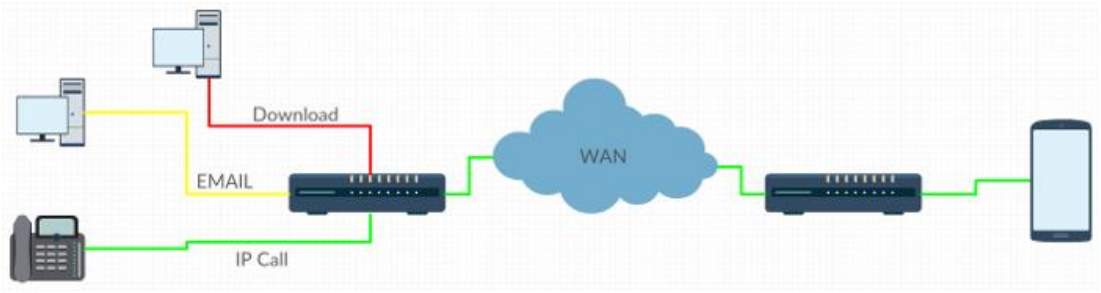
Nämä bitit määrittävät tarkemmin mitä valvontaohjelmisto piirtää graafisesti, jolloin on mahdollista nähdä, jos pakettien tippuminen aiheuttaa viivettä puheluissa jotka kulkevat IP:n päällä. Lisäksi tutkittiin, onko kuvaajalle mahdollista piirtää kaksi käyrää vertailtavaksi. Näissä käyrissä on saman tukiaseman kaksi eri QoS-määrittelyä. Tämän lisäksi piti tutkia onko olemassa vaihtoehtoisia ohjelmistoja, jos nykyinen ohjelmisto ei toimintoa tukenut.

2 QUALITY OF SERVICE

2.1 Yleisesti

Quality of Service kuvaa tietoverkon kykyä pitää yllä tarjoamansa palvelun tasoa. Sen päätarkoituksena on kontrolloida yhteyksien kaistanleveyttä niin, että priorisoitu liikenne toimii aina mahdollisimman hyvin. Sen lisäksi se kontrolloi viivettä (*latency*) ja sen vaihtelua (*jitter*) sekä paranneltua hävikin tunnistamista (*improved loss characteristics*) (Cisco 2012).

QoS käyttää tekniikkaa, jossa liikenne merkataan datapakettiin käyttämällä esimerkiksi Class of Service:ä (CoS) tai Differentiated Services Code Point:a (DSCP). Erona näillä on eri verkkokerroksen (*network layer*) käyttö datan merkkauksessa (Froehlich 2016). Kaikki data lähtee ulospäin paketteina, jotka QoS-palvelu merkkaa. Vastaanottava pää, tai vaihtoehtoisesti välissä oleva linkki, ymmärtää priorisoida kaistan QoS:n merkkaamille paketeille. Tilannetta on havainnollistettu kuvassa 3.1.



Kuva 2.1: *Reitittimellä on vikatilanteen vuoksi rajoitettu kaista käytössä, jolloin se priorisoi puhelun läpipääsyn.*

Verkkoliikenteen priorisointi riippuu kohteesta. Esimerkiksi yritys, joka käyttää IP-puhelimia, haluaa priorisoida VOIP-puhelut (*voice over ip*) sähköpostia tai http-protokollaa korkeammalle. IP-puhelimella tarkoitetaan puhelimia, jotka ovat suoraan yhteydessä internettiin, eivätkä käytä perinteistä puhelinlinjaa. Ilman priorisointia puhelut voivat alkaa kokea pakettien hukkumista, koska esimerkiksi selainta voidaan käyttää samalla johonkin paljon verkkokaistaa tarvitsevaan. Tällöin verkkokaistan kapasiteetti ei enää riitä sekä puhelun että selaimen tarpeisiin. Puhelun osapuolille tämä osoittautuu puheen pätkimisenä.

2.2 Quality of Service käytännössä

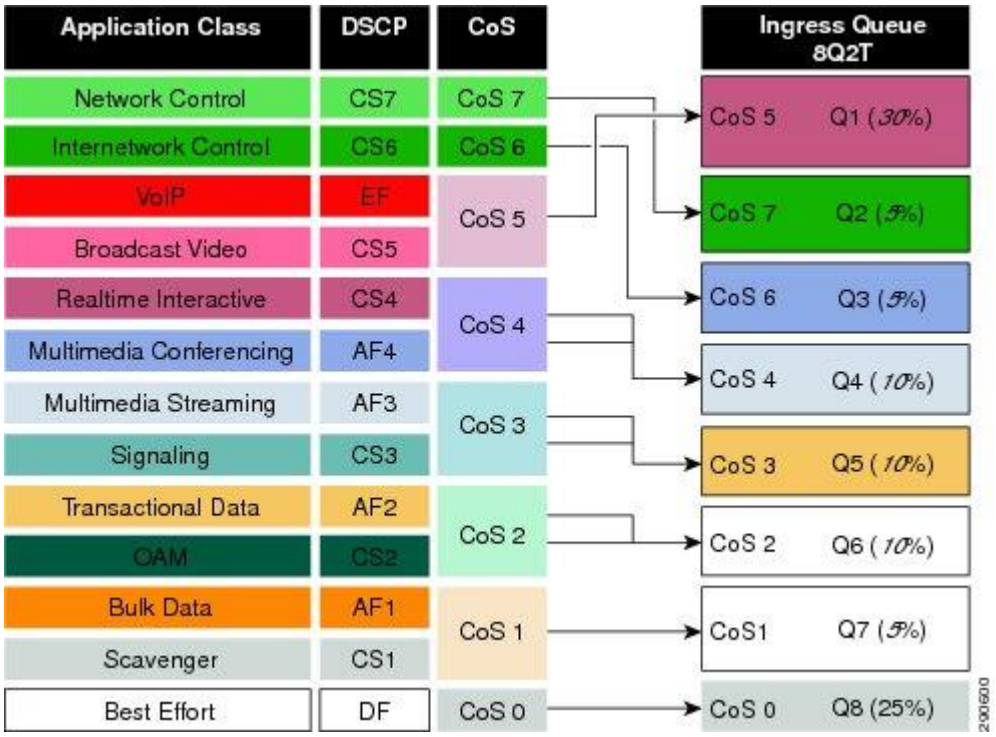
Helpoin esimerkki ovat puhelut, jotka kulkevat IP:n päällä, mutta myös videotoisto-palvelut kuten YouTube tai Netflix vaativat oman QoS määrittelyn. YouTube ja Netflix ovat tällä hetkellä suurimmat kaistaa vaativat palvelut joihin teleoperaattoreiden pitää kiinnittää huomiota (Raikisto 11.2016). Tämän lisäksi IoT:ssa (*Intenet of Things*), jossa informaatio kulkee reaaliajassa, pienetkin viiveet tai pakettien putoamiset saattavat pahimmassa tapauksessa, esimerkiksi tehtaan tuotantolinjalla, aiheuttaa merkittäviä taloudellisia tappioita (Froehlich 2016).

2.3 CoS ja DSCP

2.3.1 CoS

CoS eli Class of Service toimii OSI-mallin kakkoskerroksessa ja tukee vain ethernet-ympäristöä jossa virtuaaliset lähiverkot ovat käytössä (IEEE 802.1Q). Sen toimintaperiaatteena on ryhmitellä tietynlainen liikenne yhteen ja kohdella niitä jokaista omana ennalta määriteltynä luokkanaan joilla on oma prioriteettitasonsa (Rouse 2008). CoS käyttää kolmea bittiä ethernet kehyksessä jonne merkataan kunkin kehyksen luokka. Kehysluokkia on yhteensä kahdeksan (0-7) ja näistä nolla on varattu Best-Effort-luokalle. Best-Effort-luokka on oletusluokka (Burke 2010).

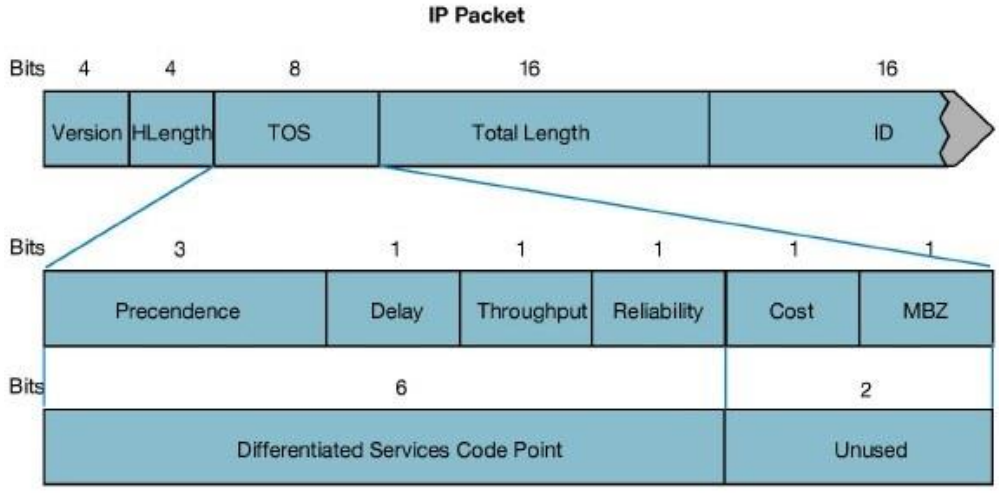
Best-Effort:illa tarkoitetaan luokkaa, joka ei anna sata prosenttista varmuutta käyttäjän datan toimittamisesta eteenpäin, eikä sillä myöskään ole prioriteettiluokkaa. Näin ollen toimitusaika on kiinni kaistan leveydestä (Wikipedia 2016). Lisäksi nykyaikaiset verkkolaitteet voivat myös kääntää CoS-tunnisteet DSCP-luokkatunnisteiksi. Tämä parantaa suurempien verkkojen toimintaa huomattavasti (Burke).



Kuva 2.2: *Sovellutuksen luokan määräytyminen (Cisco Networks 2013).*

2.3.2 DSCP

Koska CoS:in käyttämät 3 bittiä eivät anna mahdollisuuksia liikenteen tarkempaan priorisointiin, IETF (*Internet Engineering Task Force)* on kehittänyt uudenlaisen palvelun *Differentiated Services*. DiffServ käyttää kuutta bittiä IPv4-paketin ToS-kentästä ja mahdollistaa huomattavasti monipuolisemman merkintätavan.



Kuva 2.3: *DiffServ käyttää kuutta bittiä ToS -kentästä, joka on IPv4-paketin sisällä (Sonicwall).*

DiffServ:iä käytettäessä paketille annetaan käytännössä yksi edelleentoimituskäytännöistä. Edelleentoimituskäytännön mahdollisuuksia on yhteensä 64. Näitä käytäntöjä kutsutaan *per hop behaviours*(*PHB)* (Rouse 2005). DiffServ toimii verkossa niin, että sille luodaan oma toimialue, jonka sisällä kaikki laitteet käyttävät samaa palvelukäytäntöä ja PHB:tä (Huawei 2016). DiffServ käyttää pakettien merkkaamiseen normaalisti neljää *Assured Forwarding* luokkaa joissa on kolme eri tasoa. Nämä tasot määrittävät, koska paketti pudotetaan (RFC2597 1999). Asiaa voisi ajatella näin: paketti on auto, joka on määrätty tietylle moottoritielle (AF-luokka) ja tällä tiellä sillä on etuajo-oikeus perustuen siihen onko se paketti-, henkilö- vai urheiluauto. Kun tietyö poistaa osan kaistoista ja tie alkaa tukkeutua, pakettiautoja ei enää päästetä läpi.

AF-määrittelyn tarkoitus on estää pitkäaikaista ruuhkaa luokkien sisällä, mutta silti sallia hetkelliset ruuhkat jotka johtuvat satunnaisista pakettirykelmistä. Pakettirykelmä tarkoittaa, että laitteelle saapuu suuri määrä paketteja samaan aikaan. Tähän käytetään yleensä aktiivista jononhallinta-algoritmiä kuten esimerkiksi *Random Early Drop (RED)* (RFC2597).

	Assured Forwarding (AF) behavior group			
	Class 1	**Class 2**	**Class 3**	**Class 4**
Low drop probability	AF11 (DSCP 10)	AF21 (DSCP 18)	AF31 (DSCP 26)	AF41 (DSCP 34)
Med drop probability	AF12 (DSCP 12)	AF22 (DSCP 20)	AF32 (DSCP 28)	AF42 (DSCP 36)
High drop probability	AF13 (DSCP 14)	AF23 (DSCP 22)	AF33 (DSCP 30)	AF43 (DSCP 38)

Taulukko 1: *AF luokkien jaottelu.*

Vaikka DiffServ:ä käytetään nykyään paljon, se on taaksepäin yhteensopiva ja tukee *Class Selector*:ia. Tämä tarkoittaa sitä, että se tukee datapaketteja, jotka on merkannut laite, joka ei käytä DiffServ:ä. Ennen DiffServ:ä IPv4-verkoissa käytettiin ns. Etuajo-oikeutta joka määritettiin paketin Type of Service-kentässä. Vaikkei tämä ole ollut laajalti käytössä, IETF päätti säilyttää yhteensopivuuden, joten DiffServ määrittää myös Class Selector PHB:n (Wikipedia 2017).

AF-luokkien lisäksi DiffServ käyttää kahta muuta edelleenlähetystekniikkaa. Näistä ensimmäinen on *Default Forwarding*, joka on käytännössä Best-Effort. Default Forwarding käsittelee kaikki paketit, joilla ei ole erillistä määrittelyä. Toinen edelleenlähetystekniikka on *Expedited Forwarding*, jolla haetaan alhaista viivettä, pakettihävikkiä ja viiveen vaihtelua. Tämä tekniikka saa aina korkeimman prioriteetin ja sitä käytetään äänen ja videon lähettämisessä (Wikipedia).

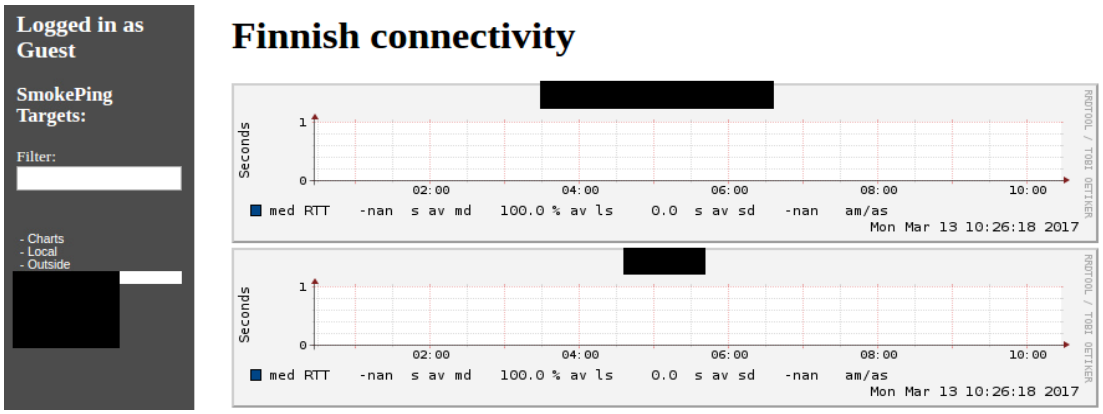
2.4 Smokeping

Smokeping on viiveseurantatyökalu. Se lähettää paketteja ennalta määritettyihin kohteisiin ja laskee, kauanko niillä kestää matkustaa sinne ja takaisin. Sille on kerrottu kuinka usein, ja kuinka monta pakettia lähetetään. Kun kohteilta saadaan vastaus, Smokeping laskee viiveen mediaanin ja sen, kuinka monta pakettia on matkalla hukattu. Tämän jälkeen Smokeping piirtää näistä tiedoista graafisen esityksen jota verkonvalvoja voi tarkastella. Graafinen esitys helpottaa verkonvalvojan työtä havaita ongelmatilanteet. Pakettien tippuminen tai verkon ylikuormittuminen eivät ole suotuisia asioita, ja tämän työkalun tarkoitus on kertoa mahdollisimman nopeasti verkonvalvojalle, jos jotain on vialla (Smokeping 2014).

3 TESTIYMPÄRISTÖ

3.1 Konfigurointi

Vaikka toimeksiantaja halusi työn lopputuloksen itselleen vain teoriana, on työhön sisällytetty myös pienimuotoinen käytännön työ, josta saataisin dokumentaatiota oppilaitokselle ja itse opinnäytetyöhön.
Työ aloitettiin asentamalla Linux Mint käyttöjärjestelmä yhdelle PC:lle. Sen päälle asennettiin LAMP (*Linux, Apache, mySQL, php*). LAMP toimii web-serverinä Smokeping-ohjelmistoa varten. Smokeping piirtää tiedot reaaliajassa suoraan verkkosivulle, josta ne voi käydä lukemassa mistä tahansa. Se tukee myös kirjautumista, mutta työn helpottamiseksi tätä ominaisuutta ei käytetty.



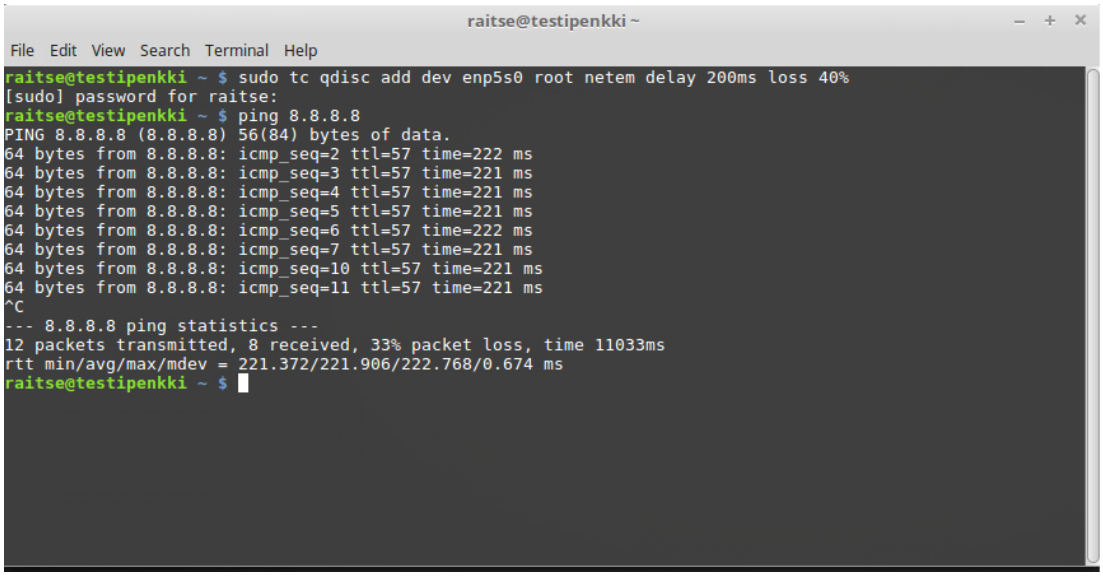
Kuva 3.1: *Smokepingiin voi lisäsä useita eri kohteita joita se "pingaa".*

Smokeping on tarkalleen ottaen ohjelmakirjasto, joka koostuu useammasta pienestä Linux -ohjelmasta joita käyttäjä konfiguroi mielensä mukaan. Kaikkia osia ei tarvitse kuitenkaan käyttää, vaan käyttäjä saa itse päättää, mitkä osat asennetaan. Koe vaati ICMP (Internet Control Message Protocol) pakettien lähettämistä, joten aluksi selvitettiin, mitkä luotaimet (Probe) pitää olla asennettuna. Fping-niminen luotain osoittautui parhaaksi vaihtoehdoksi. Se tukee Type of Service-bittien lähettämistä, ja on viimeiseksi päivitetty, joten se otettiin käyttöön ja konfiguroitiin (Liite 1).

Pakettiliikenteen seuraamista varten asennettiin Wireshark-ohjelma, jonka avulla nähdään sisältääkö Smokeping:in lähettämät paketit halutut tiedot, eli tässä tapauksessa Type of Service-bitit. Wireshark näkee kaikkien tietokoneesta lähtevien, ja saapuvien

pakettien tiedot. Se on kaikille saatavilla oleva, mutta verkonvalvojille suunnattu työkalu, jota käytetään yleensä vianetsinnässä.

Linux tarjoaa automaattisesti Windows -ympäristöä paremmat työkalut, joiden avulla voidaan simuloida mahdollisimman todenmukainen tilanne, jossa esiintyy viivettä, sekä pakettien tippumista (Packet Loss), joita Smokeping haluttiin piirtämään. Komento *"tc qdisc add dev enp5s0 root netem delay 200ms loss 40%"* luo simulaation enp5s0 verkkosovittimelle 200 millisekunnin viiveestä, sekä tiputtaa 40% kaikista paketeista.

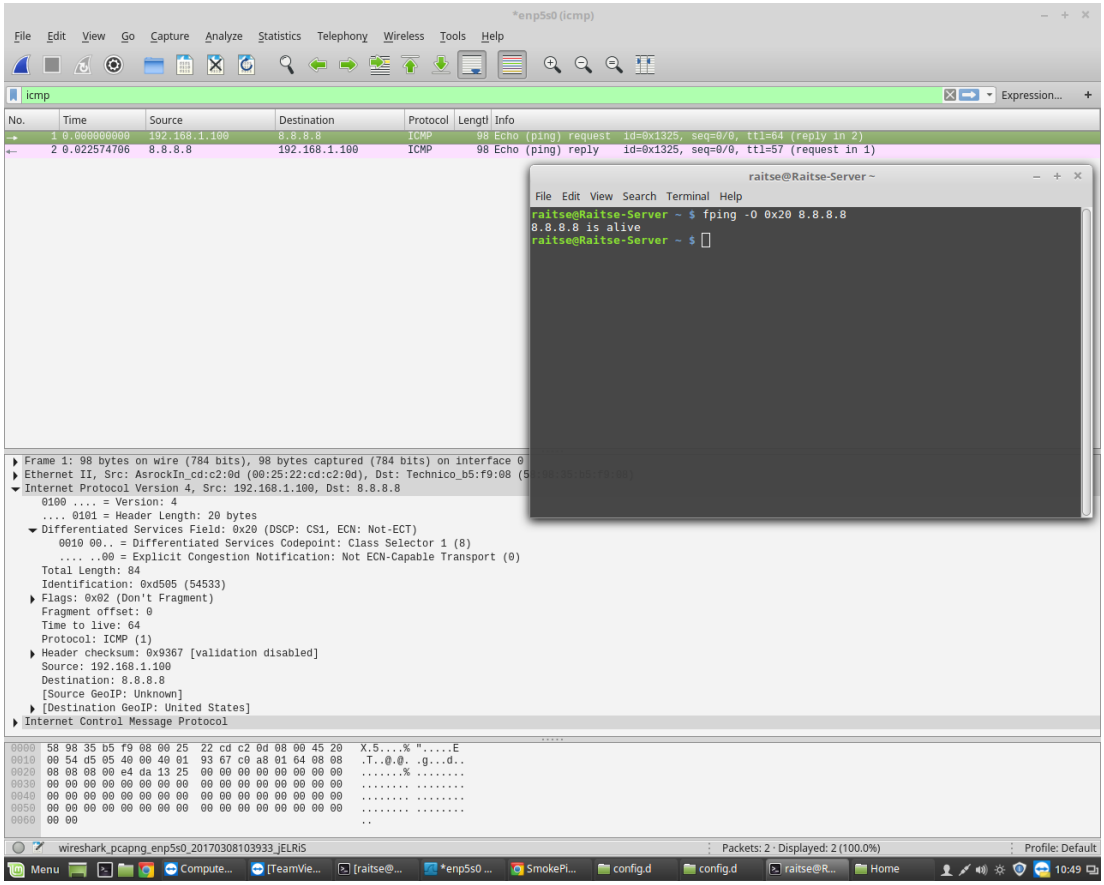


Kuva 3.2: *Linux tukee natiivisti viiveen ja pakettien tippumisen simuloimista.*

3.2 Testaus

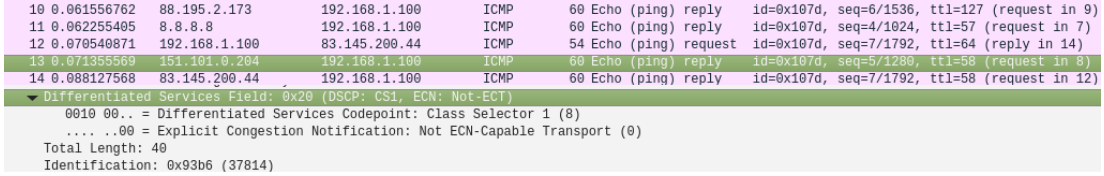
Testiympäristössä hälytyksien luominen vaati aikaa, koska paketteja oli lähetettävä ulkopuolisille tahoille ja näin ollen vältettävä niiden lähettämistä liian nopeasti. Myös Smokeping:in konfiguraation saaminen toimivaan tilaan vaati työtä, koska sen dokumentaatio ei ollut kovin selkeä tai laaja. Kokeen aikana jouduttiin useaan kertaan tukeutumaan kolmannen osapuolen sivustoihin esimerkiksi luotaimen konfiguroinnin aikana esintyneeseen ongelmaan, jossa testiympäristön käyttämän internet-yhteyden palveluntarjoaja filtteröi kaiken Smokeping:in luoman liikenteen.

Ennen varsinaista käyttöönottoa haluttiin testata, että Fping todella osaa lähettää ToS-bitit ja ne myös pääsevät ulos. Ping-komento lähetettiin Googlen nimipalvelimelle 8.8.8.8 ja siihen lisättiin ToS-biteiksi 0x20 heksadesimaali, joka tarkoittaa Class Selector 1:tä, jota voitaisiin käyttää esim. YouTuben, pelaamisen tai P2P-siirron kanssa.



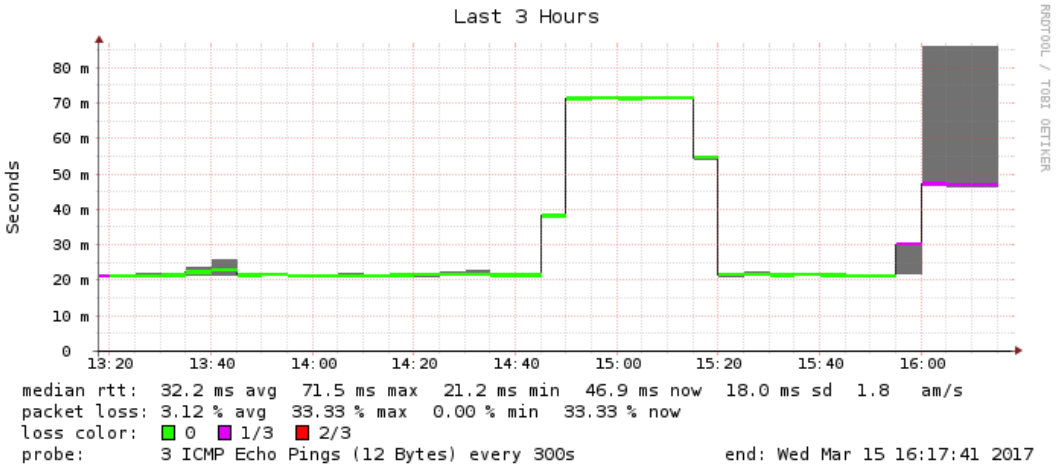
Kuva 3.3: *Kuvan keskiosassa näenme, että DiffServ -kentässä näkyvät haluamamme ToS -bitit.*

Onnistuneen testin jälkeen siirryttiin varsinaiseen työhön, eli saada Fping-luotain toimimaan Smokepingin kanssa. Luotaimen kohteiksi valittiin Googlen nimipalvelin 8.8.8.8, satunnaisia porilaisia yrityksiä, sekä testiympäristön muut laitteet. Palvelunestohyökkäyksen (*Distributed Denial of Service*) välttämiseksi paketteja lähetettiin kolme aina joka kolmassadas sekunti. Yksittäisen paketin koko oli 12 tavua.



Kuva 3.4: *Vihreällä pohjalla näkyvistä valinnoista huomaamme, että kohteeseen lähti paketti, joka sisälsi halutut ToS -bitit.*

Simuloitua viivettä luomalla Smokeping saatiin piirtämään sivuille sekä vihreää oletuskäyrää, että antamaan hälytyksen pakettien tippumisen vuoksi. Tippumisen määrää vaihdeltiin 0 ja 40% välillä, ja viivettä 0 ja 80 millisekunnin välillä. Kuva 4.5 havainnollistaa, miten viive piirtyy. Lila ja punainen merkkaavat pakettien tippumista. Korkeammalle piirretty käyrä taas kertoo viiveen nousemisesta.

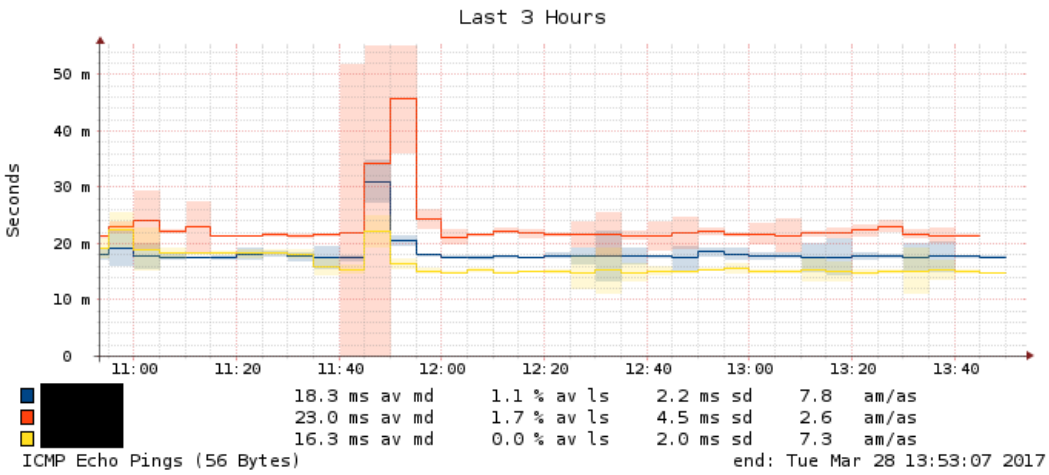


Kuva 3.5: *Smokeping kertoo yksityiskohtaisesti, paljonko viivettä on ja paljonko paketteja tippuu.*

Toimeksiantajan pyynnöstä testattiin myös, onko Smokepingin mahdollista lähettää ping-komennot samaan kohteeseen kahdella eri luotaimella. Nämä luotaimet sisältäisivät eri konfiguraation, ja tässä tapauksessa toinen toimisi oletusasetuksilla, mutta toinen sisältäisi ToS-bitit. Tämän avulla toimeksiantaja voisi paremmin vertailla tuloksia, ja nähdä tapahtuuko pakettien hukkuminen vain dataliikenteessä, vai vaikuttaako se myös puheluihin.

Tutkimus osoitti, että Smokepingin oletuskonfiguraatiolla tämä ei ole mahdollista. Lähin vastaava tilanne saatiin, kun luotiin kaksi ping-kohdetta, joiden konfiguraatiot ovat identtiset, lukuunottamatta käytettävää luotainta, joista toiseen on määritelty ToS-bitit, toiseen taas ei. Tässä tapauksessa ne saadaan piirtymään yhdelle ruudulle. Työssä käytettiin kahta Suomessa, ja yhtä Yhdysvalloissa sijaitsevaa kohdetta. Tämä tehtiin, jotta olisi mahdollista saada eriävät tulokset jokaiseen kohteeseen ja mahdollisesti saada viivettä tai pakettien hukkumista. Simuloinnin käyttämistä tässä kohdassa ei voitu käyttää, koska kaikki kohteet antaisivat identtisen tuloksen ja se hankaloittaisi tulosten

tutkimista. Kuva 4.6 havainnollistaa tätä tilannetta. Punaisella näkyvä kohde sijaitsee yhdysvalloissa.



Kuva 3.6: *Normaalissa tilanteessa Yhdysvaltojen ja Suomen välillä ei juuri ole eroa.*

Testiympäristössä tämäkaltaisen konfiguraation käyttö ei luonut mitään ongelmia, mutta toimeksiantajalla on tuhansia kohteita joita monitoroidaan. Luomalla jokaisesta kohteesta kaksi eri versiota, saattaa aiheuttaa verkolle ja palvelimelle huomattavasti rasitusta. Kohteet myös luodaan automaatiolla, jota ei testiympäristössä käytetty. Liite 2 sisältää esimerkkikonfiguraation monen kohteen piirtämisestä yhdelle ruudulle. Smokeping sisältää master/slave ominaisuuden, jossa käytetään useampaa palvelinta mittausten saamiseksi. Tällä tekniikalla voi olla mahdollista helpottaa palvelimeen kohdistuvaa rasitusta.

4 YHTEENVETO

Työn tarkoituksena oli tutkia, saako Smokeping:in lähettämiin paketteihin lisättyä ToS-bitit, ja tulokset osoittivat, että tämä on mahdollista Fping-luotaimen avulla. Testiympäristön ja varsinaisen ympäristön suurimmat erot ovat laitekannan koko, käytetyt laitteet sekä niiden eriävä konfigurointi verrattuna testiympäristöön. Tästä johtuen tämän työn tulokset saattavat muuttua, jos toimeksiantaja aikoo asiaa viedä eteenpäin. Varsinkin usean kohteen pingaus, ja tulosten piirto yhdelle ruudulle saattaa aiheuttaa verkolle rasitusta. Luotaimen vaihto vaatii ilman automaatiota huomattavan verran työtä, mutta muuten sen kanssa ei oletettavasti tule ongelmia. Testiympäristössä tämä osoittautui olevan helposti tehtävissä, joskin se vei aikaa.

Työn aikana sattuneet ongelmat osoittivat, että toimeksiantajan kannattaa harkita muihin vaihtoehtoihin siirtymistä lähitulevaisuudessa. Perustan väitteeni sille, että Smokeping on kankea ja varsinkin usean laitteen kanssa melko raskas käytettävä ja mahdollisen tuen saaminen ongelmatilanteissa on hyvin hidasta, ja todennäköisesti vastauksen antaa joku muu, kuin itse ohjelmiston ylläpitäjä. Sivuilla olevat ohjeet ovat tarkoitettu lähinnä ohjelmiston asentamiseen ja oletusasetuksien laittamiseen. Muu apu ohjeistetaan hakemaan käyttämällä postilistaa, jossa käyttäjät auttavat toinen toisiaan, tai käyttämällä arkistoja, joista kumpikin vaikutti suljetulta.

Smokeping on myös vanha. Viimeisin päivitys on vuodelta 2014, ja vaikka sivuilla mainitaan uuden version kehityksestä Github-ympäristössä, on sielläkin viimeisin päivitys tapahtunut vuonna 2013. Tietotekniikka kehittyy valtavaa vauhtia, joten neljävuotias ohjelmisto, jolla ei näytä olevan tasaista päivitysaikataulua on auttamatta jäämässä jälkeen. Vaihtoehtoisia ja uudempia ohjelmistoja löytyy, mutta niiden rajaaminen, asentaminen ja testaus vievät huomattavasti aikaa, ja tätä työtä ei ole mitoitettu niin laajaksi. Suosittelisin silti toimeksiantajaa ainakin testaamaan muita vaihtoehtoja aiemmin mainitsemistani ongelmista johtuen.

LÄHTEET

Burke, J. 2010. *CoS/QoS Basics: Understanding class and quality of service for WANs*. http://searchenterprisewan.techtarget.com/tutorial/CoS-QoS-basics-Understanding-class-and-quality-of-service-for-WANs [2017, 02/13].

Cisco Systems. 2013. *Borderless Campus 1.0 Design Guide*. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/Borderless_Campus_Network_1-0/Borderless_Campus_1-0_Design_Guide/BN_Campus_QoS.html [2017, 01/22].

Cisco Systems. 2012. *Quality of Service Networking*. http://docwiki.cisco.com/wiki/Quality_of_Service_Networking [2017, 03/29].

Froehlich, A. 2016. *The Basics Of QoS*. http://www.networkcomputing.com/networking/basics-qos/402199215 [2017, 02/12].

Heinanen, J., Baker, F., Weiss, W. & Wroclawski, J. 1999, *RFC 2597*, IETF.org.

Huawei. 2012, *Configuration Guide - QoS*, Huawei.

Institute of Electrical and Electronics Engineers, Inc 2016. *802.1Q-2014 - Bridges and Bridged Networks*. http://www.ieee802.org/1/pages/802.1Q-2014.html [2017, 01/26].

Oetiker, T. 2014. *Smokeping*. http://oss.oetiker.ch/smokeping/doc/reading.en.html [2017, 03/04].

Raikisto, V. 2017, *Henkilökohtainen tiedonanto*.

Rouse, M. 2008. *Class of Service (CoS)*. http://searchtelecom.techtarget.com/definition/Class-of-Service [2017, 02/12].

Rouse, M. 2005, , *Differentiated Services (Diffserv, or DS)*. Available: http://whatis.techtarget.com/definition/Differentiated-Services-DiffServ-or-DS [2017, 03/03].

Sonicwall 2016. *QoS Mapping*. http://help.sonicwall.com/help/sw/eng/5610/25/9/0/content/Firewall_Managing_QoS/Firewall_qosSettings.htm [2017, 02/18].

Wikipedia 2017. *Differentiated Services*. https://en.wikipedia.org/wiki/Differentiated_services [2017, 02/22].

Wikipedia 2016. *Best-effort delivery*. https://en.wikipedia.org/wiki/Best-effort_delivery [2017, 03/02].

```
// Luotaimen konfiguraatio. Vaihtoehtoja on enemmän, mutta nämä ovat tarvittavat asiat.

+ FPing

pings = 3 (Kuinka monta pakettia lähetetään)

tos = 0x20 (ToS bitti joka liitetään pakettiin heksadesimaalina)

// Kohteet joita tullaan luotaamaan.

+ Local

menu = Local
title = Local Network

++ LocalMachine

menu = Local Machine
title = This host
host = localhost
#alerts = someloss,startloss,bigloss

+++ Router

probe = Fping
menu = Router
title = Router
host = 192.168.1.1
alerts = someloss,startloss,bigloss

+++ Desktop

probe = Fping
menu = Desktop
title = Desktop
host = 192.168.1.122
alerts = someloss,startloss,bigloss

+ Outside
```

```
menu = Outside
title = Outside world
#alerts = someloss

++ Finland

menu = Finland
title = Finnish connectivity
#alerts = someloss

+++ Pori

menu = Pori
title = City of Pori
#alerts = someloss

++++ Puuvilla

probe = Fping
menu = Puuvilla
title = Kauppakeskus Puuvilla
host = porinpuuvilla.fi
alerts = someloss,lossdetect,startloss,rttdetect

++++ PlugIT

probe = FPing
menu = Plugit
title = Plugit
host = www.plugit.fi
alerts = someloss,lossdetect,startloss,rttdetect

++++ Basware

probe = FPing
menu = Basware
title = Basware
host = www.basware.fi
alerts = someloss,lossdetect,startloss,rttdetect

++ Google

probe = FPing
menu = Google
```

```
title = Google connectivity
host = 8.8.8.8
alerts = someloss,lossdetect,startloss,rttdetect
```

```
*** Probes ***

+ FPing

binary = /usr/bin/fping

++ FPingNormal

++ FPingCoS
tos = 0x68                                  // Käytetään luokkaa AF31

*** Targets ***

+ World

++ GoogleCoS

probe = FPingCoS
menu = Google
title = Google connectivity with CoS
host = 8.8.8.8
alerts = someloss,lossdetect,startloss,rttdetect

++ GoogleNormal

probe = FPingNormal
menu = Google
title = Normal Google connectivity
host = 8.8.8.8
alerts = someloss,lossdetect,startloss,rttdetect
```

++ MultiHost

menu = Both Probes
title = Google CoS Comparison
host = /World/GoogleCoS \
/World/GoogleNormal